

Amendments to the claims:

1. (Currently amended) A method for navigating between two or more application programs, each application program capable of being instantiated to form ~~a~~ an application program instance, ~~said the~~ method comprising ~~the steps of~~:
 - (a) embedding ~~and an~~ enabling engine in an origin application program and instantiating the origin application program to provide an origin application program instance;
 - (b) invoking the enabling engine for the origin application program instance, ~~which wherein the enabling engine is responsive to the origin application program instance for enabling navigation~~;
 - (c) interrogating a rule-base and retrieving ~~one or more~~ at least one condition ~~conditions associated with the origin application program instance~~;
 - (d) utilizing ~~said the~~ at least one condition ~~one or more conditions to query and evaluate data managed by the origin application program instance; and~~
 - (e) establishing and displaying ~~one or more~~ at least one navigation path ~~paths to a user for navigating to at least one target application program, which is different than the origin application program~~;
where the at least one navigation path is displayed via a graphical display by utilizing results of said evaluation the queried and evaluated data.
2. (Currently amended) The method ~~of for navigating between two or more programs as claimed in Claim 1, the method further comprising the step of~~:
enabling ~~said the~~ user to select ~~selection of a navigation path from said the one or more at least one navigation path paths thus displayed.~~
3. (Currently amended) The method ~~for navigating between two or more programs as claimed in of Claim 1, wherein said the rule-base includes at least one action or more actions associated with said the origin application program instance, said the method further comprising the step of~~:

responsive to the user selection of the navigation path, instantiating the a target application program to provide a target application program instance, which is a function of the user selection of the navigation path and navigating to the target application program instance by utilizing the at least one action of said one or more actions.

4. (Currently amended) The method ~~for navigating between two or more programs as claimed in of~~ Claim 1, ~~the method~~ further comprising the steps of:
storing and retaining a navigation path associated with the origin application program instance.

5. (Currently amended) The method ~~for navigating between two or more programs as claimed in of~~ Claim 4, ~~the method~~ further comprising the step of:
enabling the user to navigate backwards from the target application program instance to the origin application program instance by displaying the navigation path associated with the origin application program instance via ~~said the~~ graphical display.

6. (Currently amended) The method ~~for navigating between two or more programs as claimed in of~~ Claim 1, wherein the steps step of establishing and displaying one or more the at least one navigation path paths further comprises comprising:
utilizing state data managed by the origin program instance and role of said user for evaluating the at least one or more navigation path paths available to the user based on (a) state data managed by the origin application program instance and (b) a role of the user.

7. (Currently amended) The method ~~for navigating between two or more programs as claimed in of~~ Claim 3, further comprising:
performing remaining actions from ~~said the at least one action or more actions that are not~~ utilized for executing functionality supported by the target application program instance.

8. (Currently amended) The method ~~for navigating between two or more programs as claimed in~~ of Claim 1, wherein the target application program ~~is~~ comprises a web browser.

9. (Currently amended) The method ~~for navigating between two or more programs as claimed in~~ of Claim 1, wherein the rule-base resides on a network.

10. (Currently amended) The method ~~for navigating between two or more programs as claimed in~~ of Claim 1, wherein the rule-base resides locally on the user's system.

11. (Currently amended) The method ~~for navigating between two or more programs as claimed in~~ of Claim 1, wherein the target application program resides locally on the user's system.

12. (Currently amended) The method ~~for navigating between two or more programs as claimed in~~ of Claim 1, wherein the target application program resides on a network.

13. (Currently amended) The method ~~for navigating between two or more programs as claimed in~~ of Claim 1, wherein the at least one navigation path ~~paths, which are established and displayed comprises~~ comprise a workflow for sequentially performing ~~one or more~~ tasks.

14. (Currently amended) ~~The method for navigating between two or more application programs as claimed in~~ of Claim 7, wherein the functionality supported by the target application program ~~is~~ comprises displaying a particular web page.

15. (Currently amended) A system for navigating between two or more application programs, each program capable of being instantiated to form a program instance having data associated therewith, ~~said the~~ system comprising:

(a) a rule-base for storing at least one condition ~~or more conditions and one or more actions~~ associated with ~~said~~ an origin application program instance;

(b) an enabling engine embedded in ~~said the~~ origin application program instance and responsive to the origin application program instance ~~enabled for interrogating said the rule-~~ base and retrieving ~~said the at least one or more conditions condition and said one or more~~ actions;

(c) ~~said the~~ enabling engine including a mechanism for utilizing the at least one condition to querying said and evaluate data managed by the origin application program instance data, evaluating said data against said one or more conditions, and, utilizing the queried and evaluated data, establish ~~establishing at least one or more navigation path paths for navigating to~~ at least one target application program, which is different than ~~the origin application program~~; and

(d) a display means for displaying ~~said the at least one or more navigation path~~ paths to a user.

16. (Currently amended) The system ~~for navigating between two or more programs as claimed in of~~ Claim 15, wherein ~~said the display means enables said the user to select selection of~~ a navigation path from ~~said the at least one or more navigation path paths thus~~ displayed.

17. (Currently amended) The system ~~for navigating between two or more programs as claimed in of~~ Claim 15, wherein ~~said the rule-base includes at least one action associated with the origin application program instance, and the~~ enabling engine further comprises:

means for instantiating a target application program, ~~which is a function of to provide a target application program instance, responsive to the~~ user selection of the navigation path; and

means for navigating to the target application program instance by utilizing the at least one action of said one or more actions.

18. (Currently amended) ~~The system for navigating between two or more programs as claimed in of~~ Claim 15, the system further comprising:

~~a~~ means for storing and retaining a navigation path associated with the origin application program instance.

19. (Currently amended) ~~The system for navigating between two or more programs as claimed in of~~ Claim ~~18~~ 17, the system further comprising:

means for enabling the user to navigate backwards from the target application program instance to ~~the an~~ origin application program instance by displaying the navigation path associated with the origin application program instance via ~~said the display means~~.

20. (Currently amended) ~~The system for navigating between two or more programs as claimed in of~~ Claim 15, the system further comprising:

means for evaluating the at least one navigation path based on (a) means for ~~utilizing~~ state data managed by the origin application program instance and (b) a role of said the ~~user for evaluating one or more navigation paths available to the user.~~

21. (Currently amended) ~~The system for navigating between two or more programs as claimed in of~~ Claim 17, the system further comprising:

means for performing remaining actions from ~~said the~~ at least one action or more ~~actions that are not yet performed~~ utilized for executing functionality supported by the target application program instance.

22. (Currently amended) ~~The system for navigating between two or more programs as claimed in of~~ Claim 17, wherein the target program comprises a web browser.

23. (Currently amended) ~~The system for navigating between two or more programs as claimed in of~~ Claim 15, wherein the rule-base resides on a network.

24. (Currently amended) ~~The system for navigating between two or more programs as claimed in~~ of Claim 15, wherein the rule-base resides locally on a user's computer system.

25. (Currently amended) ~~The system for navigating between two or more programs as claimed in~~ of Claim 15, wherein the target application program resides locally on a user's computer system.

26. (Currently amended) ~~The system for navigating between two or more programs as claimed in~~ of Claim 15, wherein the target application program resides on a network.

27. (Currently amended) ~~The system for navigating between two or more programs as claimed in~~ of Claim 15, wherein the at least one navigation path ~~paths comprise~~ comprises a workflow for sequentially performing ~~one or more~~ tasks.

28. (Currently amended) ~~The system for navigating between two or more programs as claimed in~~ of Claim 21, wherein the functionality supported by the target application program ~~is comprises~~ displaying a particular web page.

29. (Cancelled)

30. (Cancelled)

31. (Currently amended) A program storage device readable by a machine, tangibly embodying a program of instructions executable by the machine to perform a method ~~steps for~~ navigating between two or more application programs, each application program capable of being instantiated to form a an application program instance, ~~said the~~ method steps comprising:

(a) embedding an ~~and~~ enabling engine in an origin application program and instantiating the origin application program to provide an origin application program instance;

(b) invoking the enabling engine for the origin application program instance, ~~which wherein the enabling engine is responsive to the origin application program instance for~~ enabling navigation;

(c) interrogating a rule-base and retrieving at least one or more conditions condition associated with the origin application program instance;

(d) utilizing ~~said the at least one condition or more conditions~~ to query and evaluate data managed by the origin application program instance; and

(e) establishing and displaying at least one or more navigation path paths to a user for navigating to at least one target application program, which is different than the origin application program;

where the at least one navigation path is displayed via a graphical display by utilizing the queried and evaluated data results of said evaluation.

32. (Currently amended) The program storage device ~~as claimed in of~~ Claim 31, wherein the method further comprises comprising the step of:

enabling said the user to select selection of a navigation path from said the at least one or more navigation path paths thus displayed.

33. (Currently amended) The program storage device ~~as claimed in of~~ Claim 31, wherein ~~said the~~ rule-base includes at least one action or more actions associated with ~~said the~~ origin application program instance, and the method further comprises comprising the step of:

responsive to the user selection of the navigation path, instantiating a target application program, which is a function of to provide a target application program instance -the user selection of the navigation path and navigating to the target application program instance by utilizing the at least one action of said one or more actions.

34. (Currently amended) The program storage device ~~as claimed in of~~ Claim 31, wherein the method further comprises comprising the step of:

storing and retaining a navigation path associated with the origin application program instance.

35. (Currently amended) ~~The program storage device as claimed in of Claim 34 31, wherein the method further comprises comprising the step of:~~

enabling the user to navigate backwards from the target application program instance to the origin application program instance by displaying the navigation path associated with the origin application program instance via ~~said the~~ graphical display.

36. (Currently amended) ~~The program storage device as claimed in of Claim 31 35, wherein the step steps of establishing and displaying the at least one or more navigation path paths further comprises comprising:~~

~~utilizing state data managed by the origin program instance and role of said user for evaluating the at least one or more navigation path paths available to the user based on (a) state data managed by the origin application program instance and (b) a role of the user.~~

37. (Currently amended) ~~The program storage device as claimed in of Claim 33, wherein the method further comprises comprising the step of:~~

performing remaining actions from ~~said the at least one action or more actions that are not utilized yet performed~~ for executing functionality supported by the target application program instance.

38. (Currently amended) ~~The program storage device as claimed in of Claim 31, wherein the target application program is comprises~~ a web browser.

39. (Currently amended) ~~The program storage device as claimed in of Claim 31, wherein the rule-base resides on a network.~~

40. (Currently amended) ~~The program storage device as claimed in of Claim 31, wherein the rule-base resides locally on the user's system.~~

41. (Currently amended) ~~The program storage device as claimed in of Claim 31, wherein the target application program resides locally on the user's system.~~

42. (Currently amended) The program storage device ~~as claimed in of~~ Claim 31, wherein the target application program resides on a network.

43. (Currently amended) The program storage device ~~as claimed in of~~ Claim 31, wherein the at least navigation path ~~paths that are established and displayed comprise~~ comprises a workflow for sequentially performing ~~one or more~~ tasks.

44. (Currently amended) The program storage device ~~as claimed in of~~ Claim 37, wherein the functionality supported by the target application program ~~is~~ comprises displaying a particular web page.